A TRIANGULATION APPROACH TO ASSYMPOTICALLY EXACT CONDITIONS FOR FUZZY SUMMATIONS

A. Kruszewski(1), A. Sala(2), T.M. Guerra(3), C. Ariño(4)

(1) LAGIS UMR CNRS 8146, Ecole Centrale de Lille, Cité Scientifique, Villeneuve d’asq 59655, France
(2) Instituto de Automática e Informática Industrial, Dept. of Systems Engineering & Control, Universidad Politécnica de Valencia, E-46022 Spain
(3) LAMIH UMR CNRS 8530, Université de Valenciennes et du Hainaut-Cambrésis, Le Mont Houy, 59313 CEDEX 9, France
(4) Dept. Industrial Systems Engineering and design, Universitat Jaume I Av. Vincent Sos Baynat, s/n. 12071 Castelló de la Plana, Spain

Abstract: Many Takagi-Sugeno fuzzy control-synthesis problems in literature are expressed as the problem of finding decision variables in a double convex sum (fuzzy summation) of positive definite matrices. Matrices’ coefficients in the summation take values in the standard simplex. This paper presents a triangulation approach to the problem of generating simplicial partitions of the standard simplex, in order to set up a family of sufficient conditions and, in parallel, another family of necessary ones for fuzzy summations. The conditions proposed in this paper are asymptotically exact as the size of the involved simplices decreases: its conservativeness vanishes for a sufficiently fine partition (sufficiently dense mesh of vertex points). The set of conditions is in the form of linear matrix inequalities, for which efficient software is available.

Keywords: Linear matrix inequality, conservatism reduction, nonlinear models, Takagi-Sugeno models, fuzzy control.

1. INTRODUCTION

It is well known that nonlinear dynamic systems fulfilling some continuity and sector conditions can be (at least locally) expressed as fuzzy combinations of linear models.
following the sector-nonlinearity technique in [1]; the obtained models are denoted as Takagi-Sugeno (TS) fuzzy systems expressed in the form:

\[ \dot{x}(t) = \sum_{i=1}^{r} h_i(z(t)) \left( A_i x(t) + B_i u(t) \right) \]

with input \( u(t) \in \mathbb{R}^m \), state \( x(t) \in \mathbb{R}^q \), vertex model matrices \( A_i, B_i \), and interpolating membership functions \( h_i \). Functions \( h_i \) always take values in the so-called standard simplex (i.e., they are positive and \( \sum_{i=1}^{r} h_i = 1 \)). When a controller \( u(t) = \sum_{i=1}^{r} h_i(z(t)) L_i x(t) \) is put in place, the closed loop takes the form of an homogeneous double sum

\[ \dot{x}(t) = \sum_{i=1}^{r} \sum_{j=1}^{r} h_i(z(t)) h_j(z(t)) G_{ij} x(t) , \text{ with } G_{ij} = A_i + B_i L_j , \]

and so do stabilization and performance conditions based on quadratic Lyapunov functions (see [1, 2, 3] for details on TS modeling and controller design). Such conditions usually lead to the problem of proving positive definiteness of a homogeneous matrix in the form of a double fuzzy summation \( \sum_{i=1}^{r} \sum_{j=1}^{r} h_i h_j Q_{ij} \). Each of the terms \( Q_{ij} \) in the summations is usually a linear expression of some matrix unknowns.

For instance, the above appears in polytopic gain scheduling control design [4], uncertain models using parameter-dependent Lyapunov functions [5] and fuzzy control of nonlinear systems [1]. Early results used a quadratic Lyapunov function, but piecewise analysis [6,7] as well as non-quadratic approaches [8, 9] have also been developed, for instance, in the fuzzy control case. In general, some of the above-referenced control problems may lead to higher-dimensional summations (having more
than two summation indices: $\sum_{i=1}^{r} \ldots \sum_{j=1}^{r} h_i \ldots h_j Q_{i,j}$. For example, non quadratic approaches for stabilization may lead to triple summations in the LPV case [10], the TS case [8], and also for TS static output feedback [11].

Checking positive-definiteness of the expression $\sum_{i=1}^{r} \sum_{j=1}^{r} h_i h_j Q_{i,j}$ is known to be a co-NP problem, referred in the literature as the matrix copositivity problem [12]. There exist well-known relaxations which are computationally tractable and provide sufficient conditions for the problem under study [1, 13, 14]. Some of them are asymptotically exact as complexity increases [15], i.e., sufficient conditions become also necessary when the complexity parameter tends to infinity. The result of such relaxations in the discussed control problems is expressed as a linear matrix inequality (LMI) problem [16] which can be efficiently solved by commercially available software. However, the non-fulfillment of the above referred sufficient conditions leads to the question of whether the control problem is either unfeasible (say, a TS fuzzy model cannot be stabilized with the correspondent choices of controller structure and Lyapunov function) or, on the contrary, it will be feasible if the sufficient conditions are made less conservative by increasing their complexity [15]. To the authors’ knowledge, this is an open question.

The objective of this paper is closing that question by providing two sets of LMI conditions:

(a) a family of sufficient conditions which are less conservative as its complexity increases (solving, in an alternative way, the same problem as [15]), and
(b) necessary conditions which are also less conservative as its complexity increases.

It will be proved that both sets of conditions are asymptotically exact (their conservativeness tends to zero as complexity tends to infinity): if a problem is either (strictly) feasible or unfeasible it can be proved in a finite number of complexity-augmentation steps. Unfortunately, as the underlying copositiveness problem is computationally hard, the number of required steps as well as the amount of computing resources depend on the problem conditioning, which is unknown beforehand. This paper closes the aforementioned problem from a theoretical point of view, but its practical application is limited by the available computational capabilities.

Sufficient conditions in [15] are based on Polya’s theorems. The alternative ones offered in this paper are based on triangulation (or, more in general, simplicial partitions in higher dimensions). Also, taking advantage of triangulation, the necessary conditions in this work exploit the fact that positive-definiteness of the summations for arbitrarily chosen values of the membership functions $h_i$ is a necessary condition for the problem to be feasible (conveniently, the chosen points are the vertices of the simplices of the current mesh). The complexity-augmentation steps are set up by splitting some of the simplices to form a finer mesh.

The structure of the paper is the following. The next section states the problem, introduces notation, and reviews previous results from the literature on Takagi-Sugeno fuzzy control design. Using a complexity parameter related to the maximum size of a family of sets, Section 3 proposes sufficient conditions (asymptotically exact as the size
parameter tends to zero) and necessary ones (also asymptotically exact) for the fuzzy summation problem. Section 4 uses such results to propose algorithms, and Section 5 presents numerical examples of its use and compare them to previous results. A conclusion section closes the paper.

2. PRELIMINARIES

2.1 Takagi-Sugeno fuzzy systems

Many nonlinear systems can be expressed as a convex interpolation (fuzzy summation) of linear ones. For instance, consider the system \( \dot{x}(t) = \sin(x(t)) \cdot x(t) \). Evidently, as \( \sin(x(t)) \) always takes values in \([-1, 1]\), the expression for the state derivative may be expressed as a nonlinear interpolation between two linear systems, \((-1) \times x(t)\) and \((+1) \times x(t)\), i.e., the model can be rewritten as \( \dot{x} = h_1(x(t)) \cdot x(t) + h_2(x(t)) \cdot (-x(t)) \)

with \( h_1 = (1 + \sin(x)) / 2 = 1 - h_2 \geq 0 \) so that \( h_1 + h_2 = 1 \). Functions \( h_1 \) and \( h_2 \) are denoted as membership functions. The generalization of this idea gives rise to the so-called sector-non-linearity technique for fuzzy modeling [1]: if \( f(x) \) is bounded in a region of interest \( \Omega \), i.e., \( \alpha \leq f(x) \leq \beta \) then \( f(x) \) can be expressed as \( h_i(x) \alpha + (1 - h_i(x)) \beta \).

The idea can be easily extended to \( f(x) \) being a matrix-valued function and \( x \) a vector. Then, a wide class of nonlinear systems \( \dot{x} = f(x)x + g(x)u \) may be expressed as a TS fuzzy system \( \dot{x}(t) = \sum_{i=1}^{r} h_i(z(t))(A_i x(t) + B_i u(t)) \), with \( h_i(x) \geq 0 \) and \( \sum_{i=1}^{r} h_i = 1 \).

2.2 Fuzzy controllers and fuzzy summations
As stated in the introduction, Lyapunov-based conditions for stability / stabilization / estimation with some performance measures (decay rate, $H_\infty$ attenuation, etc.) of Takagi-Sugeno fuzzy systems are often expressed in the form:

$$\sum_{i=1}^{r} \sum_{j=1}^{r} h_i(z) h_j(z) Q_{ij} > 0$$

where $Q_{ij}$ are matrices formed with plant model parameters and decision variables, being the latter related to Lyapunov functions and controller parameters [1]. The integer $r$ denotes the number of “linear models” or “fuzzy rules”. Expression (1) will be denoted as “double fuzzy summation”. Some notation and examples will be now introduced.

Let us denote as $\Delta$ the $(r-1)$-dimensional standard simplex whose $r$ vertex coordinates are the canonical basis of $\mathbb{R}^r$:

$$e_i^T = \begin{bmatrix} 0 & \ldots & 0 & 1 \text{ row} & 0 & \ldots & 0 \end{bmatrix}$$

so that the column vector $H$ formed by stacking the membership functions is:

$$H = \sum_{i=1}^{r} h_i e_i ,$$

with $\sum_{i=1}^{r} h_i = 1$ and $h_i \geq 0$. Hence, $H$ belongs to $\Delta$. On the following, a point in $\Delta$, i.e., a particular feasible value of the membership functions, will be denoted by its coordinate vector $H$.

For instance, one of the simplest problems to which (1) applies is the state-feedback stabilization of a TS model [17]:

$$\dot{x}(t) = \sum_{i=1}^{r} h_i (z(t))(A_i x(t) + B_i u(t))$$

(3)
with a parallel distributed compensation (PDC) [2] control law:

\[ u(t) = -\sum_{i} h_i(z(t))L_i x(t) \]  

(4)

where matrices \( L_i \) are the controller gains to be found.

Indeed, conditions for the decreasing of quadratic Lyapunov function \( V(x) = x(t)^T P x(t) \) give the following well-known result in [1]:

**Theorem 1:** Closed-loop TS model (3) under control law (4) is globally asymptotically stable if there exist matrices \( X > 0 \) and \( M_i, i \in \{1, \ldots, r\} \), such that for all \( z(t) \) in a particular premise set:

\[ \sum_{j=1}^{r} \sum_{i=1}^{r} h_i(z(t))h_j(z(t))Q_{ij} > 0 \]

(5)

with \( Q_{ij} = -A_iX + B_iM_j + M_j^TB_i^T - XA_i^T \). Moreover if (5) is satisfied, the control gains are given by \( L_i = M_iX^{-1} \).

Introducing performance requirements results in different expressions for \( Q_{ij} \), see [1,18,19]. For example, controllers achieving disturbance rejection (bounding the \( L_2 \rightarrow L_2 \) induced norm between disturbances and output) can be obtained by using in (1), the following expression for \( Q_{ij} \) [19]:

\[
Q_{ij} = -\begin{bmatrix}
XA_i^T + A_iX + R_{ui}^T B_{aj}^T + B_{ui} R_{aj} & B_{ui} & XC_i^T + R_{ui}^T D_{wi}^T \\
B_{wij}^T & -\gamma I & D_{wij}^T \\
C_iX + D_{wij} R_{aj} & D_{wij} & -\gamma I
\end{bmatrix}
\]

where \( \gamma \) is the desired closed-loop norm bound, \( A_i, B_{aj}, B_{ui}, C_i, D_{ui} \) and \( D_{wij} \) are TS model matrices and \( R_{ui}, X \) are decision variables (see [19] for details).

2.3 Positive-definiteness Conditions for fuzzy summations
Proving (1) amounts to proving positive-definiteness of a matrix which depends on a set of premise variables \( z \) via the nonlinear functions \( h_i(.) \). In TS fuzzy models, the premise variables \( z \) may include some (or all) of the system state variables.

Without loss of generality, symmetry of \( Q_i \) and a convex sum condition \( \sum_{i=1}^{r} h_i(z) = 1 \), \( h_i(z) \geq 0 \), are assumed to hold.

In most of the positivity conditions found in the literature, there is no assumption on \( h_i(z(t)) \) except that of the convex sum property. In this way, any function \( h_i(z(t)) \) in (5) can be replaced with scalars, and positiveness is required to hold for all their possible values in the standard simplex, yielding the condition:

\[
\Theta(H) = \sum_{j=1}^{r} \sum_{i=1}^{r} h_j Q_i > 0 \quad \forall H = \begin{bmatrix} h_1 \\ \vdots \\ h_r \end{bmatrix} \in \Delta \tag{6}
\]

**Remark 1**: Requiring the positive-definiteness condition (6) to hold “for all” possible values of the memberships in the standard simplex is usually conservative with respect to the original problem (1) when \( z \) depends on some of the state variables (this is the shape-independence issue discussed in [20]). The reader is referred to [21, 22] for some options to reduce such conservativeness. In general, problem (6) is very conservative regarding the stability of a nonlinear system, even if exactly modeled by a TS fuzzy model with the sector-nonlinearity technique [1]. Apart from the conservativeness of (6) with respect to (1), the latter is in itself conservative. Indeed, the choices of a Lyapunov function family and a particular controller structure add conservativeness to the shape-independence one. From the three main sources of conservatism in fuzzy control
(Lyapunov function choice, copositivity in fuzzy summations, disregarding of membership function shape as discussed in [20]), only the second one is addressed in this paper: its objective is to find the less conservative conditions to solve (6).

In order to ensure that (6) holds, several sufficient conditions have been proposed in literature. The first and most evident is $Q_{ij} > 0$, $i, j \in \{1, \ldots, r\}$. Indeed, if all $Q_{ij} > 0$, then (6) is a linear combination of positive definite matrices with non-negative coefficients. Closely related is the following lemma from [1]:

**Lemma 1:** Expression (6) holds if for $i, j \in \{1, \ldots, r\}$:

\[
Q_{ii} > 0 \quad (7) \\
Q_{ij} + Q_{ji} \geq 0 \quad j < i \quad (8)
\]

More relaxed (i.e., less conservative) sufficient conditions have been later proposed [13, 19, 14], even asymptotically exact ones [15] as complexity increases. In the examples in section 5, the following lemmas from [19] and [13], recalled here for convenience, will be used:

**Lemma 2 [19]:** Expression (6) holds if for $i, j \in \{1, \ldots, r\}$:

\[
Q_{ii} > 0, \quad (9) \\
\frac{2}{r-1}Q_{ii} + Q_{ij} + Q_{ji} \geq 0 \quad j \neq i \quad (10)
\]

**Lemma 3 [13]:** Expression (6) holds if there exist some slack matrices $\Theta_{ij} = (\Theta_{ji})^T$, $i, j \in \{1, \ldots, r\}$ such that:

\[
Q_{ii} \geq \Theta_{ii} \quad (11)
\]
\[ Q_{ij} Q_{ji} \geq \Theta_{ij} + \Theta_{ji} \quad j > i \]  \hspace{1cm} (12)

\[ \Theta = \begin{bmatrix} \Theta_{11} & \cdots & \Theta_{1r} \\ \vdots & \ddots & \vdots \\ \Theta_{r1} & \cdots & \Theta_{rr} \end{bmatrix} > 0 \]  \hspace{1cm} (13)

Lemmas 2 and 3, widely cited, offer a quite reasonable compromise between computational complexity and accuracy. Improved results in [14], and further improvements in [15], are less conservative but more computationally demanding.

Next section will propose an alternative to [15] to solve the fuzzy summation problem (6) based on applying lemma 1 to a mesh in the standard simplex. It also gives conditions for (6) not to hold (which is not carried out in [15]). Note, however, that the results in the next section will be proved valid for any of the relaxations of lemma 1 in literature, such as lemmas 2, 3 and all cases in [15], which includes [13] and [14] as particular cases (see corollary in Section 3.1).

3. MAIN RESULTS

The lemmas and literature references in the previous section are only sufficient conditions for positive-definiteness of the left hand side of (6), i.e., their non-fulfillment does not imply that (6) is unfeasible.

This section proposes modifications to the above settings, based on a mesh in the standard simplex, so that:

(a) a set of sufficient conditions is provided (theorem 3)

(b) a set of related necessary conditions is also provided (theorem 4)

(c) both sets of conditions reduce its conservativeness as points are added to the mesh and,
(d) when the mesh becomes dense enough (in formal terms, as the size of the elements of its triangulation -simplicial partition- tends to zero), they become non-conservative, i.e., they are “asymptotically exact” in the sense stated in the theorems below.

In this way, conclusive answers to the feasibility or unfeasibility of the problem (6) may be obtained (the computational cost, however, depends on the particular problem, see the example section). The basic idea is realizing that, as (6) must hold at the simplex vertices, condition (7) is necessary for (6).

**Remark on notation:** It is easy to realize that any LMI positive-definiteness conditions for (6) found in literature must include (7), either explicitly or implicitly via some conditions which entail (7).

For convenience, when considering a particular relaxation in literature, let us denote the left-hand side of all the other additional positive-definiteness conditions to (7) as a set of matrices: \( R(\Delta) = \{ \Gamma_\lambda, \lambda \in \{1, \ldots, n_\lambda\} \} \), \( n_\lambda \) being the number of additional conditions.

For instance, for lemma 2, we have:

\[
R(\Delta) = \left\{ \frac{1}{r-1} Q_{11} + Q_{12} + Q_{21}, \ldots, \frac{1}{r-1} Q_{11} + Q_{1r} + Q_{r1}, \frac{1}{r-1} Q_{22} + Q_{21}, Q_{12}, \ldots \right\}
\]

so the matrices \( \Gamma_\lambda \) result from arbitrarily denoting sequentially each condition as

\[
\Gamma_1 = \frac{1}{r-1} Q_{11} + Q_{12} + Q_{21}, \text{ and so on.}
\]

For lemma 3, taking into account that (11) and (13) imply (7), we have:

\[
R(\Delta) = \left\{ Q_{21} + Q_{12} - \Theta_{21}, \ldots, Q_{1r} - \Theta_{r1}, Q_{r1} - \Theta_{1r}, \ldots, \Theta \right\}
\]

and the \( \Gamma_\lambda \) are suitably formed by enumeration.
The main idea behind the argumentations below is to set up a partition of the standard simplex $\Delta$, formed by $n$ simplicial subsets $\Delta^{(k)}$, $k \in \{1, \ldots, n\}$, i.e., fulfilling $\Delta^{(1)} \cup \ldots \cup \Delta^{(n)} = \Delta$. Each simplicial subset $\Delta^{(k)}$ is defined by $r$ vertices: $e_i^{(k)} \in \Delta^{(k)}$, $i \in \{1, \ldots, r\}$ whose canonical coordinates in the standard simplex will be denoted by $\alpha_i^{(k)}$, i.e., $e_i^{(k)} = \left[ \alpha_1^{(k)} \ldots \alpha_r^{(k)} \right]^T$, and

$$e_j^{(k)} = \sum_{j=1}^r \alpha_{ij}^{(k)} e_j$$

(14)

Obviously $\Delta^{(k)}$ are convex sets, and the sum of the canonical coordinates of each vertex is one because all vertices belong to $\Delta$.

The partition condition ensures that any point $H \in \Delta$ belongs to at least one partition subset $\Delta^{(k)}$. From its definition in (2), the membership function values are the canonical coordinates of $H$. Let us now define the coordinates of $H$ in $\Delta^{(k)}$ (i.e., in the basis formed by the vertices of $\Delta^{(k)}$) as the unique $H^{(k)} = \left[ \begin{array}{c} h_{1}^{(k)} \\ \vdots \\ h_{r}^{(k)} \end{array} \right]$, so that the equation below holds:

$$H = \sum_{i=1}^r h_i^{(k)} e_i^{(k)}$$

Then, using (14):

$$H = \sum_{j=1}^r \sum_{i=1}^r h_i^{(k)} \alpha_{ij}^{(k)} e_j$$

(16)

So, finally, we can write a coordinate transformation between $\Delta^{(k)}$ and $\Delta$ as follows

$$h_j = \sum_{i=1}^r h_i^{(k)} \alpha_{ij}^{(k)}$$

in matrix form, can be expressed as:

$$H = \Omega^{(k)} H^{(k)}$$

(17)
Note that feasibility of the original problem (6) is equivalent to simultaneous feasibility in all the partition simplices, so (6) can be equivalently stated as requiring the double sum positiveness to hold for all $H$ in $\Delta^{(k)}$, for all $k$. After applying the above coordinate change to each simplex, the result given in the following theorem follows straightforwardly from the discussion above.

**Theorem 2:** Given a simplicial partition of $\Delta$, composed by $n$ subsets $\Delta^{(k)}$ such that

$$\bigcup_k \Delta^{(k)} = \Delta,$$

necessary and sufficient conditions to (6) are:

$$\forall k \in \{1, \ldots, n\} \left\{ \forall H \in \Delta^{(k)} \sum_{i=1}^{r} \sum_{j=1}^{r} h_i^{(k)} h_j^{(k)} Q_{ij}^{(k)} > 0 \right\}$$

(18)

where the $\alpha_j^{(k)}$ are the coordinates of the vertices $e_i^{(k)}$, $i \in \{1, \ldots, r\}$ of $\Delta^{(k)}$.

### 3.1. SUFFICIENT CONDITIONS.

Once problem (6) is restated as the equivalent problem (18), using any sufficient condition in Section 2.1 on the $n$ double sums of (18) (i.e., one for each partition set) will give sufficient LMI conditions for (6). In this way the original condition has been replaced by multiple ones on smaller simplexes.

**Asymptotical exactness of the sufficient conditions.** We will proceed now to prove that the conservativeness vanishes as the maximum size of the subsets $\Delta^{(k)}$ decreases, i.e., the conditions become necessary and sufficient (exact) in the limit case. The result will hold irrespective of the sufficient conditions chosen from the previously discussed alternatives in literature for each subset $\Delta^{(k)}$ in (18).
Consider a generic algorithm which, for any finite value of an integer complexity parameter $n$, generates a partitioning of the standard simplex in $n$ simplicial subsets $\Delta^{(k)}$, $k \in \{1, \ldots, n\}$, so that $\bigcup_{k=1}^{n} \Delta^{(k)} = \Delta$. The vertices of each subset $\Delta^{(k)}$ are denoted by $e^{(k)}_i$, $i \in \{1, \ldots, r\}$. Some possibilities for such an algorithm are considered in Section 4.

Consider the element-wise maximum norm for vectors and matrices. The proposition below describes how problem (18) appears in the limit case as the partition grows finer, so the maximum size of the sets tends to zero.

**Proposition 1.** If the above partition-generation algorithm fulfills

$$\lim_{n \to \infty} \left( \max_{i,j,1 \leq k \leq n} \left( \left\| e^{(k)}_i - e^{(k)}_j \right\| \right) \right) = 0,$$

then $\lim_{n \to \infty} \left( \left\| Q^{(k)}_j - Q^{(k)}_i \right\| \right) = 0$ in (18).

**Proof:** Indeed, when the size of the simplicial subsets in the partition tends to zero, we can write: $\left\| e^{(k)}_j - e^{(k)}_j \right\| = \left\| \sum_{m=1}^{r} \left( \alpha_{im}^{(k)} - \alpha_{jm}^{(k)} \right) e_m \right\| \to 0$ so that $\alpha_{im}^{(k)} - \alpha_{jm}^{(k)} \to 0$ for all $m$. From the definition of $Q^{(k)}_j$ in (18), we have $Q^{(k)}_j - Q^{(k)}_i = \sum_{j=1}^{r} \sum_{m=1}^{r} \alpha_{jm}^{(k)} \left( \alpha_{jm}^{(k)} - \alpha_{jm}^{(k)} \right) \times Q_{im}$ and the desired result follows straightforwardly.

**Theorem 3:** Choose any positive-definiteness condition for each double sum (18) in $\Delta^{(k)}$, denoted as $Q^{(k)}_j > 0$, $R\left( \Delta^{(k)} \right) = \{ \Gamma^{(k)}_\lambda, \lambda \in \{1, \ldots, n_\lambda\} \}$, $n_\lambda$ being the number of LMI conditions in the chosen approach. Consider a partitioning algorithm of the standard
simplex $\Delta$, in $n$ simplicial subsets $\Delta^{(k)}$, $k \in \{1, \ldots, n\}$, so $\bigcup_{k=1}^{n} \Delta^{(k)} = \Delta$ with vertices $e_i^{(k)}$, $i \in \{1, \ldots, r\}$ such that $\lim_{n \to \infty} \left( \max_{i,j,k \leq n} \left\| e_i^{(k)} - e_j^{(k)} \right\| \right) = 0$. Then,

(a) the conditions:

$$\forall k \in \{1, \ldots, n\}, \left\{ Q_u^{(k)} > 0 \quad \forall i \in \{1, \ldots, r\} \right\} \quad \left\{ \Gamma_{\lambda \lambda}^{(k)} > 0 \quad \forall \lambda \in \{1, \ldots, n_\lambda\} \right\}$$

are sufficient for (6) to hold, and

(b) there exists a sufficiently large $n$ such that they are necessary and sufficient (i.e., they are asymptotically exact).

**Proof:** As stated before, (6) holds if and only if (18) holds. Hence, necessary conditions to (18) are:

$$\forall k \in \{1, \ldots, n\}, \forall i \in \{1, \ldots, r\}, \; Q_u^{(k)} > 0, \quad \lim_{n \to \infty} \left( \max_{i,j,k \leq n} \left\| e_i^{(k)} - e_j^{(k)} \right\| \right) = 0,$$

According to proposition 1 and the fact that $\lim_{n \to \infty} \left( \max_{i,j,k \leq n} \left\| e_i^{(k)} - e_j^{(k)} \right\| \right) = 0$, $\lim_{n \to \infty} \left\| Q_y^{(k)} - Q_u^{(k)} \right\| = 0$. A trivial sufficient condition to ensure $\sum_{j=1}^{r} \sum_{i=1}^{r} h_j^{(k)} h_j^{(k)} Q_y^{(k)} > 0$ in (18) is $Q_y^{(k)} > 0$ $\forall k \in \{1, \ldots, n\}$, $\forall i,j \in \{1, \ldots, r\}$. Note that $Q_y^{(k)} > 0$ can be written as $Q_y^{(k)} + \left( Q_y^{(k)} - Q_u^{(k)} \right) > 0$.

Since $\lim_{n \to \infty} \left\| Q_y^{(k)} - Q_u^{(k)} \right\| = 0$, there exist a sufficiently large $n$ such that condition (21) implies (20), as by continuity and compactness argumentations, there exist $\varepsilon > 0$ such that $Q_{ii}^{(k)} \varepsilon$ and there exists $n$ such that $\left\| Q_{\{ij\}}^{(k)} - Q_{\{ii\}}^{(k)} \right\| \varepsilon$. This proves that sufficient conditions $Q_y^{(k)} > 0$ become also necessary for a sufficiently large $n$. □
Remark 2: It is straightforward to prove that any relaxation from literature can be used as well, and it will require a lower or equal $n$. Indeed, any sufficient condition for the simplices in Theorem 2 provides, evidently, sufficient conditions for (6). Interestingly, asymptotic exactness also holds: if $Q_y^{(k)}>0$ for all $i$, $j$ and $k$ then all conditions in literature discussed in Section 2.1 hold. Hence, proving asymptotical exactness for $Q_y^{(k)}>0$ automatically proves such exactness for any set of less conservative sufficient conditions chosen to ensure $\sum_{i=1}^{r} \sum_{j=1}^{r} h_i^{(k)} h_j^{(k)} Q_y^{(k)}>0$ in (18), like for lemmas 1, 2, 3, [14], [15], etc.

Also, the partition condition (disjoint interior of any two sets $\Delta^{(k)}$) stated in the theorem is not actually needed, but only a “cover” with a family of decreasing-size simplices. Nevertheless, the latter is quite a sensible choice, and as such it will be used in the examples.

3.2. NECESSARY CONDITIONS

Note now that checking (6) in a finite set of points in $\Delta$ is a necessary condition for feasibility. Interestingly, such finite set of points may be formed by the vertices of the simplicial partition used in the previous subsection, so the necessity check amounts to only $Q_y^{(k)}>0$. The idea is expressed as Theorem 4 below, which gives necessary and asymptotically sufficient conditions (i.e., the conditions are always necessary and become sufficient for a large enough $n$) for (18), and hence (6).
**Theorem 4:** Consider an algorithm partitioning the standard simplex $\Delta$ in $n$ simplicial subsets $\Delta^{(k)}$, $k \in \{1, \ldots, n\}$ with vertices $e_i^{(k)}$, $i \in \{1, \ldots, r\}$ such that
\[ \lim_{n \to \infty} \max_{i,j \in \Delta} \left( \|e_i^{(k)} - e_j^{(k)}\| \right) = 0. \]
Suppose there exists a point $H$ in the standard simplex so that
\[ \Theta(H) < 0 \] in (6). Then, there exists a sufficiently large $n \in \mathbb{N}$, a $k \in \{1, \ldots, n\}$ and a $i \in \{1, \ldots, r\}$ such that $Q_{ii}^{(k)} > 0$ does not hold.

**Proof:** By continuity (eigenvalues are a continuous function of the matrix coefficients), there exists a radius $\delta$, so that (6) does not hold in any of the points of an open ball $B$ with such a radius centered in $H$. Then, for a sufficiently large number of simplices in the partition, when the maximum norm between vertices is less than $\delta$, at least one of them will be in $B$, so that the corresponding condition $Q_{ii}^{(k)} > 0$ will not hold. Indeed, conditions $Q_{ii}^{(k)} > 0$ are the evaluation of $\sum_{j=1}^{r} \sum_{j=1}^{r} h_i h_j Q_{ij} > 0$ at the vertices of the simplices. □

**4. ALGORITHM PROPOSAL**

The theorems in the previous section refer to a generic partition-generating algorithm such that it generates finer partitions as a complexity parameter (the number of partition sets $n$) increases. This section discusses some sensible choices for practical implementation of such an algorithm.

Note that conditions in theorem 3 are sufficient for (6) and asymptotically exact; conversely, conditions in theorem 4 are necessary for positive definiteness of the left-
hand side in (6) and asymptotically sufficient for non-negativeness. Hence, by using both theorems 3 and 4, a suitable algorithm generating the partitions allows stating the feasibility or unfeasibility of condition (6) in a finite number of steps. The only limitation of these theorems is the LMI solver resources (software + hardware) and the time available to solve the LMI problem.

Note: The number of steps will not be finite if the problem is marginally feasible, i.e., if \( \Theta(H) \geq 0 \), because the strict inequality is required in Theorems 3 and 4.

Let us outline some algorithmic proposals implementing the above idea.

Simplicial partitions of polyhedral sets are generated from triangulation algorithms, once a set of simplicial vertices (which, obviously, must include the canonical vertices of the standard simplex) is chosen. In this paper the Delaunay \( p \)-triangulation algorithm of MATLAB is used [23].

![Fig.1: Example of the use of Delaunay triangulation.](image-url)

For example, to analyze a fuzzy system with 3 rules (see Figure 1), consider the 2-dimensional standard simplex \( \Delta \) in \( \mathbb{R}^3 \). As \( h_3 = 1 - h_1 - h_2 \) is a dependent coordinate it is not represented in the figure. Five points are defined (symbolized by \( \circ \)) and 4
tetrahedral simplices are obtained. Its projections on the plane \((h_1, h_2)\) are plotted on the referred figure.

**Remark 3:** In order to ensure \(\lim_{n \to \infty} \max_{i,j,k} \left( \left\| e_i^{(k)} - e_j^{(k)} \right\| \right) = 0\), it is sufficient to split the largest edges (i.e., those actually attaining the maximum value of the Euclidean norm \(\max_{i,j,k} \left( \left\| e_i^{(k)} - e_j^{(k)} \right\| \right)\)) by its middle point, generating two smaller simplices for the next value of \(n\).

Once the above ideas have been introduced, the following algorithm is proposed as a general scheme. It gives a solution whenever it exists; otherwise it establishes the unfeasibility (except in marginal cases). The step number is denoted by \(s\).

**Algorithm 1:**

*Initialization:* \(s = 1\) initialize the set of vertices of \(\Delta: e_i^{(k)} (s) \in \Delta_{(e)} \) corresponding to the canonical basis of \(\mathbb{R}^n\).

*Step 1:* Test the necessity conditions \(\forall k \in \{1, \ldots, n\}, \forall i \in \{1, \ldots, r\}, Q_i^{(k)} > 0\). If the LMI problem is unfeasible stop the algorithm and conclude that (6) does not hold; otherwise go to step 2.

*Step 2:* Test the sufficient conditions (19) of theorem 3. If the LMI problem is feasible stop the algorithm and conclude that (6) holds; otherwise go to step 3.
Step 3: $s := s + 1$ Compute a finer partition by introducing new vertices with the constraints $\bigcup_{k=1}^{n} \Delta_{s+1}^{(k)} = \Delta$ and:

$$\sup_{i,j,k} \left( \| e_{i}^{(k)} (s+1) - e_{i}^{(k)} (s+1) \| \right) < \sup_{i,j,k} \left( \| e_{i}^{(k)} (s) - e_{i}^{(k)} (s) \| \right)$$ 

(In our examples this step uses the Delaunay triangulation). Go to step 1.

Remark 4: As the chosen points and the Delaunay triangulation ensures

$$\lim_{n \to \infty} \left( \sup_{i,j,k} \left( \| e_{i}^{(k)} - e_{j}^{(k)} \| \right) \right) = 0,$$

the algorithm always gives a feasibility/infeasibility result, except in marginally feasible problems. Nevertheless, limited computational resources can leave some ill-conditioned strictly feasible problems undecided.

Remark 5: There exist an infinite number of possible decompositions of $\Delta$, as well as many options for the necessary or sufficient conditions at each step. Although they are asymptotically equivalent as the partition gets finer, they are not equivalent in terms of overall computing efficiency in the finite case: the computational resources needed to find a conclusive result for a particular problem might depend on such choices. This suggests that the way of choosing the decomposition of $\Delta$ must be deeper studied.

Some heuristic ways to accelerate the search of a solution can be thought of. One of them is focusing the search in the simplicial region(s) where one or several condition(s) of the LMI problem (19) fail(s).

Algorithm 2 presents a variation of the previous one, incorporating such a search criteria. Interestingly, it has been formalized using a single generalized eigenvalue
problem, joining in one step two of those in the previous algorithm (avoiding the need to solve a different LMI problem in each of the steps).

**Algorithm 2:**

*Initialization:* Fix a user-defined “splitting period” $T \in \mathbb{N}$ (applying Remark 3 in order to ensure convergence whatever the results of the heuristic searches below are). Set up an iteration count variable as $s = 1$, and initialize the set of vertices of $\Delta$: $e_i^{(k)}(s) \in \Delta(s)$ as the $n$ vectors in the canonical basis of $\mathbb{R}^n$.

**Step 1:** Solve the following generalized eigenvalue maximization problem:

$$
\begin{aligned}
\max & \quad t \\
Q_{ii}^{(k)} & > 0 \quad \forall i \in \{1,\ldots,r\} \quad \forall k \in \{1,\ldots,n\} \\
\Gamma_{ii}^{(k)} & > t \cdot I \quad \forall \lambda \in \{1,\ldots,n\} \quad \forall k \in \{1,\ldots,n\}
\end{aligned}
$$

If the above problem is unfeasible, stop the algorithm, since (6) does not hold. If there is a solution with $t > 0$ the algorithm stops since (6) holds. Otherwise (i.e., in case of feasibility with $t < 0$), go to step 2.

**Step 2:**

$s := s + 1$ Compute a smaller partition by introducing new vertices as follows:

**if** $s \text{ mod } T = 0$, split in half the largest edges in order to ensure:

$$\max_{i,j,k} (\| e_i^{(k)}(s + 1) - e_j^{(k)}(s + 1) \|) < \max_{i,j,k} (\| e_i^{(k)}(s) - e_j^{(k)}(s) \|), \quad \lim_{s \to +\infty} \left( \max_{i,j,k} (\| e_i^{(k)} - e_j^{(k)} \|) \right) = 0$$

so that convergence results in the previous sections apply

**else** (i.e. $s \text{ mod } T \neq 0$),
check a different search direction, in particular, find the simplex $k$ and the condition
\[ \lambda \in \{1, \ldots, n\} \] such that $\Gamma^{(k)}_\lambda$ has the most negative eigenvalue. Considering its vertices, $e_i^{(k)}$, add a point between any two of such vertices and generate the new partition $\Delta^{(k)}_{s+1}$ by triangulation.

end

Go to step 1.

Remark 6: The presented results can be easily extended to control problems involving fuzzy summations with greater dimensions than 2 (output feedback, dynamic controllers, etc).

Indeed, proving positive definiteness of a $p$-dimensional summation
\[ \sum_{i=1}^{r} \sum_{\ell=1}^{r} h_i \ldots h_p Q_{h_i \ldots h_p} \] in the $(p-1)$-dimensional standard simplex amounts to proving positive definiteness in each of the simplices of a partition. The above sum, restricted to simplex $\Delta^{(k)}$ is expressed as:

\[ \forall H \in \Delta^{(k)} \sum_{i=1}^{r} \sum_{\ell=1}^{r} h_i^{(k)} \ldots h_p^{(k)} Q_{h_i \ldots h_p}^{(k)} > 0, \quad Q_{h_i \ldots h_p}^{(k)} = \sum_{j=1}^{r} \sum_{\ell=1}^{r} \alpha_{h_i}^{(k)} \ldots \alpha_{h_p}^{(k)} Q_{h_i \ldots h_p}. \]

Generalized versions of theorems 3 and 4 may be given by replacing $Q_{i}^{(k)}$ by $Q_{h_i}^{(k)}$, and $Q_{j}^{(k)}$ by $Q_{h_j}^{(k)}$. Details are omitted for brevity.

5. EXAMPLES

Consider the following TS model with a free parameter $b$:
\[
\dot{x}(t) = \sum_{i=1}^{3} h_i(z(t))(A_i x(t) + B_i u(t))
\]  

(22) 

with:

\[
A_1 = \begin{bmatrix} 1.59 & -7.29 \\ 0.01 & 0 \end{bmatrix} \quad B_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad A_2 = \begin{bmatrix} 0.02 & -4.64 \\ 0.35 & 0.21 \end{bmatrix} \quad B_2 = \begin{bmatrix} 8 \\ 0 \end{bmatrix} \quad A_3 = \begin{bmatrix} -2 & -4.33 \\ 0 & 0.05 \end{bmatrix} \quad B_3 = \begin{bmatrix} -b+6 \\ -1 \end{bmatrix}
\]

and the control law \( u(t) = \sum_{i=1}^{\bar{r}} h_i(z(t))L_i x(t) \).

According to theorem 1, the following condition is sufficient to find a stabilizing PDC regulator (with a quadratic Lyapunov function) for (22):

\[
\sum_{j=1}^{\bar{r}} \sum_{i=1}^{\bar{r}} h_i(z(t)) h_j(z(t)) Q_{ij} > 0 \quad \text{with} \quad Q_{ij} = -A_i X + B_i M_j + M_j^T B_i^T - XX^T .
\]

(23) 

The efficiency of different results in the literature, as well as the ones proposed, will be compared for different values of the parameter \( b \). For instance, using the relaxation given in Lemma 1 on the system (23), the maximum value of \( b \) for which such sufficient condition is feasible (i.e., proves the existence of a stabilizing regulator) is \( \bar{b} = -1.4 \).

Let us now set up Algorithm 1 for the above problem with the following choices:

- The relaxation used for conditions (19) in Step 2 is that in Lemma 1.
- Step 3 is carried out by adding the middle point of the largest edge of all simplices \( \Delta^{(k)} \) of a partition to the list of vertices for the Delaunay triangulation. If there are several edges with the same length, the one to be split is randomly chosen.

In order to illustrate some of the algorithm characteristics, the first steps will be detailed. For instance, starting with the full simplex \( \Delta \) in \( \mathbb{R}^3 \), randomly splitting one of
the edges, results in the situation whose projection in \((h_1, h_2)\) appears in Figure 2 showing the decomposition of \(\Delta\) in 2 subsets \((\Delta^{(1)}\) and \(\Delta^{(2)}\)) using the points \(e^{(k)}_i\) (\(*\)):

\[
e^{(1)}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad e^{(1)}_2 = \begin{pmatrix} 0.5 \\ 0.5 \\ 0 \end{pmatrix}, \quad e^{(1)}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}
\]

for the first simplex, and \(e^{(2)}_1 = \begin{pmatrix} 0.5 \\ 0 \\ 1 \end{pmatrix}, \quad e^{(2)}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad e^{(2)}_3 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}\)

for the second one.

The transformed matrices \(Q^{(1)}_i, Q^{(2)}_i\) are obtained for each simplex using (18). For instance, some of the \(Q^{(k)}_i\) matrices are:

\[
Q^{(1)}_{11} = Q_{11}, \quad Q^{(1)}_{13} = Q_{13}, \quad Q^{(1)}_{31} = Q_{31}, \quad Q^{(1)}_{33} = Q_{33}
\]

\[
Q^{(1)}_{21} = 0.5\left(Q_{21} + Q_{11}\right) = -\frac{(A_1 + A_2)}{2}X + \frac{(B_1 + B_2)}{2}M_1 + (*)
\]

\[
Q^{(1)}_{12} = 0.5\left(Q_{12} + Q_{11}\right) = -A_1X + B_1\left(M_1 + M_2\right) + (*)
\]

\[
Q^{(1)}_{22} = 0.25\left(Q_{11} + Q_{12} + Q_{21} + Q_{22}\right) = -\frac{(A_1 + A_2)}{2}X + \frac{(B_1 + B_2)}{2}\left(M_1 + M_2\right) + (*)
\]

with notation \(M + (*)\) denoting \(M + M^T\) to obtain a symmetric matrix from an expression \(M\).
Fig. 2: Decomposition in 2 subsets in first iteration.

At each iteration $s$, a bisection algorithm is used to find both the maximum value $\bar{b}$ of $b$ (so that feasibility can be asserted using the sufficient conditions), and the minimum $\underline{b}$ (for the necessary conditions to hold). In this way, each partition associated to each algorithm iteration yields an interval of undecided values for $b \in [\underline{b}, \bar{b}]$, instead of only $\bar{b}$ as current literature does. The width of such interval decreases as iterations progress. Evidently, as the number of vertices increases, the number of sets generated by the triangulation algorithm also increases and, hence, so it does the computational load.

Table 1 below gives some results for different number of vertices/sets using Algorithm 1 with the proposed choices.

- The row $\bar{b}$ represents the maximum value proved to be a stabilizable model with a PDC regulator and a quadratic Lyapunov function (i.e., sufficient conditions do not hold for greater values).

- The row $\underline{b}$ represent the minimum value for which the unfeasibility of the stabilization problem can be asserted (i.e., necessary conditions for quadratic stability do not hold for lower values). Obtaining stabilizing PDC regulators using a quadratic
Lyapunov function is impossible for \( b \geq b' \). Hence, a stabilizing regulator may exist, but either it is not a PDC or it is not based on a quadratic Lyapunov function.

As expected, the algorithm increases its precision (the undecided interval shrinks) as iterations progress and vertices are added. This behavior is illustrated in Figure 3. Note that the computation time increases “heavily” with the number of vertices/sets.

Table 1: \( b \) and \( \beta \) according to the number of vertices/sets and computation time

<table>
<thead>
<tr>
<th>vertices</th>
<th>3</th>
<th>5</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>50</th>
<th>100</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>sets</td>
<td>1</td>
<td>3</td>
<td>12</td>
<td>26</td>
<td>46</td>
<td>74</td>
<td>173</td>
<td>350</td>
</tr>
<tr>
<td>( b/\beta )</td>
<td>-1.4</td>
<td>0.4</td>
<td>2.6</td>
<td>4.9</td>
<td>5.1</td>
<td>5.2</td>
<td>5.9</td>
<td>6.2</td>
</tr>
<tr>
<td>time*</td>
<td>0.04</td>
<td>0.14</td>
<td>0.96</td>
<td>3.03</td>
<td>7.2</td>
<td>15.9</td>
<td>59.4</td>
<td>230</td>
</tr>
</tbody>
</table>

Fig.3: Asymptotical convergence of the algorithm 1 as the number of vertices increases.

Table 2 considers the results using both lemmas 2 and 3 without the proposed algorithm (i.e., considering only the 3 vertices of the whole simplex). These results are compared
with algorithm 1 altogether with lemma 1 (basic conditions). The algorithm stops as soon as its result for $\bar{b}$ outperforms the solution produced from lemmas 2 and 3. Then, the corresponding number of vertices is given. It can be seen that algorithm 1 improves over previous literature results, even if equipped with a more conservative sufficient condition, at the expense of increasing the number of vertices. Of course, incorporating lemmas 2 and 3 in the algorithm, instead of lemma 1, would have led to significantly better results for the same number of vertices, particularly with Lemma 3 (details omitted for brevity).

Table 2: Computation time for different approaches giving at least the same result.

<table>
<thead>
<tr>
<th>$\bar{b}$</th>
<th>Algo 1: 5 vertices</th>
<th>Tuan et al. 2001</th>
<th>Algo 1: 11 vertices</th>
<th>Liu &amp; Zhang 2003</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>-0.16</td>
<td>4.17</td>
<td>2.9</td>
<td></td>
</tr>
<tr>
<td>0.14</td>
<td>0.08</td>
<td>1.13</td>
<td>0.07</td>
<td></td>
</tr>
</tbody>
</table>

* The computation time in tables 1 and 2 is measured only for solving the LMI problems for an unfeasible value of $b$ (feasible ones can be much faster). The LMI conditions are solved with the LMI toolbox of Matlab under default options. The employed computer was a P4 3GHz with 2GB Ram.

As shown in Table 1, the conservatism can be considerably reduced as the number of vertices increases. Nevertheless, the bottom row shows the price to pay in computing time. Note that, as stated before, any relaxation in literature can be used on each simplicial set in (18). What is expected is a tradeoff between the complexity of these
relaxations (the number of slack variables and the number of LMI conditions), and the number of vertices needed to assert feasibility or infeasibility of a particular problem.

The next comparison shows how the heuristic algorithm 2 improves over the algorithm 1. In terms of computing time, the two algorithms use the same calculations per step. The efficiency is reported in table 3. The heuristic search direction in algorithm 2 seems to be more efficient when a high number of vertices are needed.

<table>
<thead>
<tr>
<th>$b$</th>
<th>vertices needed algo 1</th>
<th>vertices needed algo 2</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>11</td>
<td>10</td>
<td>feasible</td>
</tr>
<tr>
<td>5</td>
<td>17</td>
<td>18</td>
<td>feasible</td>
</tr>
<tr>
<td>5.5</td>
<td>41</td>
<td>29</td>
<td>feasible</td>
</tr>
<tr>
<td>6</td>
<td>110</td>
<td>39</td>
<td>feasible</td>
</tr>
<tr>
<td>6.8</td>
<td>59</td>
<td>16</td>
<td>unfeasible</td>
</tr>
<tr>
<td>7</td>
<td>12</td>
<td>5</td>
<td>unfeasible</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>6</td>
<td>unfeasible</td>
</tr>
</tbody>
</table>

6. CONCLUSION

This work has presented a triangulation methodology to decide, in a finite number of steps, whether a given fuzzy control problem is strictly feasible or unfeasible (with a particular choice of controller structure and Lyapunov function). This improves over mainstream literature in two directions: on one hand, both sufficient and necessary conditions are combined instead of using only sufficient ones and, on the other hand, asymptotic exactness of both sets of conditions is achieved as the size of the sets in a simplicial partition tends to zero.

The obtained conditions are asymptotically necessary and sufficient, but the computational cost increases as the conditions precision does. The number of steps will
not be finite for marginally feasible problems. Actually, computer resources determine the precision of the achievable results in practical implementations.

Some issues need to be addressed in the future regarding the choice of mesh points, the options in the triangulation algorithm and the relaxation for the sufficient conditions to be used, in order to strike the best balance between accuracy, computing time and memory requirements.

ACKNOWLEDGEMENTS

The authors are grateful to the financial support of grants XXXXXXXXX (yours), grants DPI2008-06731-C02(-01 and -02) from the Spanish Government and grant prometeo/2008/088 from Valencia regional government.

REFERENCES


